BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

As rescanning documents *will not* correct images, please do not report the images to the Problem Image Mailbox.

Offic urop'en d s brevets

(11) Publication number : **0 556 994 A1**

(12) EUROPEAN PATENT APPLICATION

(21) Application number : **93300918.5**

(22) Date of filing : **09.02.93**

(51) Int. Cl.[5] : **G03G 15/00, H04N 1/21, G06F 3/12, G06K 15/00**

(30) Priority : **10.02.92 JP 23464/92**
30.03.92 JP 74597/92

(43) Date of publication of application :
25.08.93 Bulletin 93/34

(84) Designated Contracting States :
DE FR GB IT

(71) Applicant : **CANON KABUSHIKI KAISHA**
30-2, 3-chome, Shimomaruko, Ohta-ku
Tokyo (JP)

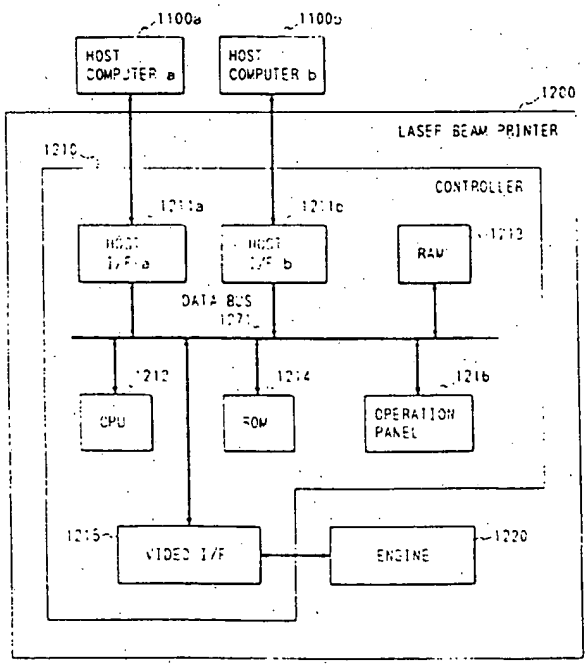
(72) Inventor : **Higuchi, Yuichi, c/o Canon Kabushiki Kaisha**
30-2, 3-chome, Shimomaruko, Ohta-ku
Tokyo (JP)
Inventor : **Sato, Nobuhiko, c/o Canon Kabushiki Kaisha**
30-2, 3-chome, Shimomaruko, Ohta-ku
Tokyo (JP)

(74) Representative : **Beresford, Keith Denis Lewis et al**
BERESFORD & Co. 2-5 Warwick Court High Holborn
London WC1R 5DJ (GB)

(54) Output method and unit.

(57) An output unit can be connected to a plurality of data generation sources. Therein, print data can be input substantially simultaneously from the respective data generation sources, and printing is performed by data generation sources. This eventually improves a working environment.

FIG. 1



EP 0 556 994 A1

BACKGROUND OF THE INVENTION

Field of the Invention

The present invention relates to an output method and unit, or more particularly, to an output method and unit enabling connection with a plurality of output data generation sources.

Related Background Art

A printer having multiple host interfaces has been introduced in the past. The respective interfaces are connected to electronic equipment (for example, host computers) for generating print data actually, whereby the operators of the electronic equipment can execute printing at their places.

However, while receiving and printing print data from, for example, a host computer A, this kind of printer does not receive or print data sent from other host computers B, C, etc.

Therefore, other operators except the one of the host computer A which is currently engaged in communication cannot print their data until the other's printing currently in progress is completed. This is discouraging in terms of a working environment.

A printer shared by multiple host computer systems (hereafter, hosts) outputs print data sent from the hosts as they are without any control.

In the aforesaid prior art, a host intending to transmit print data must issue an emulation switch instruction so that a printer to which the host is connected will be emulated to enable communication with the host. Alternatively, a user of the printer must select an emulator by performing a specified operation on the printer. This results in complex control of communication between a host and a printer or complex operations of a printer.

When a printer is shared by multiple hosts, it is hard to differentiate which printed forms are output by which hosts. The printed forms must be sorted by studying the output contents. From the viewpoint of efficiency in sorting, improvements have been awaited.

SUMMARY OF THE INVENTION

An object of the present invention is to provide an output unit capable of being connected to a plurality of data generation sources; wherein print data can be input substantially simultaneously from the respective data generation sources and printing is performed by data generation sources, whereby a working environment improves; and its control method.

Another object of the present invention is to provide an output unit comprising:

a plurality of interfaces for connecting a plurality of data generation sources;

a determining means that determines by interfaces whether a reception request is issued therefor;

an input means that receives output data via interfaces for which a reception request has been issued, and stores the received output data in one-to-one correspondence with the respective interfaces; and

a control means for controlling output processing so that output of second received data will start after output of first received data, which has been stored in the storage means, is completed.

Another object of the present invention is to provide an output method comprising:

a determining process in which it is determined by interfaces connected to a plurality of data generation sources whether a reception reques! is issued therefor;

an input process in which print data is input via interfaces for which a reception request has been issued and the input print data are stored in one-to-one correspondence with respective interfaces; and

a control process in which printing is controlled so that printing of received data, which is the second data stored in the input process, will start after printing of received dala, which is the first data stored in the input process, is completed.

Another object of the present invention is to provide an output unit capable of confirming from which data generation source output data is being received.

Another object of the present invention is to provide an output unit comprising:

a plurality of interfaces for connecting a plurality of data generation sources;

a determining means that determines by interface whether a reception request is issued therefor;

an input means that receives output data via interfaces for which a reception request has been issued, and stores the received output data in one-to-one correspondence with the respective interfaces;

a control means for controlling output processing so that output of second received data will start after output of first received data, which has been stored in the storage means, is completed; and

a notifying means that notifies externally the states of interfaces for which the determining means determines a reception request has issued or of the states of interfaces concerned with the input means.

Yet another object of the present invention is to provide an output method in which data is received via interfaces of a plurality of interfaces for which a reception request is issued, the received data are stored in one-to-one correspondence with respective interfaces, and output processing is controlled so that output of second received data will start after output of first received data, which is being stored in the storage means, is completed.

Still another object of the present invention is to provide an output unit capable of being connected to

a plurality of data generation sources, wherein output data is r ceived from the data generation sources substantially simultaneously, and output processing is executed by data gen ration sources to improve a working environment, and its control method.

Still another object of the present invention is to provide an output method in which inputs of output data of a plurality of hosts are differentiated from one another in order to simplify sorting of outputs by hosts, and an output unit in which the output method is implemented.

Still another object of the present invention is to provide an output method comprising a discrimination process in which when output data is received from a plurality of hosts and the output data is supplied, a host intending to transmit output data is discriminated from the plurality of hosts, a preparation process in which preparations are made to receive output data from the host intending to transmit output data in accordance with the result of discrimination in the discrimination process, a comparison process in which it is checked if a host which transmitted output data during the previous output is identical to the host discriminated in the discrimination process, and a control process in which output processing is controlled so that the output data will be offset in accordance with the result of comparison made in the comparison process, and an output unit in which the output method is implemented.

Still another object of the present invention is to provide an output unit that is connected to a plurality of host, receives output data from the plurality of hosts, and supplies output data, comprising a discriminating means that discriminates a host intending to transmit output data from the plurality of hosts, a preparing means that makes preparations to receive output data from a host intending to transmit output data in accordance with the result of discrimination made by the discriminating means, a comparing means that checks if a host which transmitted output data during the previous output is identical to the host discriminated by the discriminating means, and a control means that controls output so that outputs of data will be offset in accordance with the result of comparison made by the comparing means.

Still another object of the present invention is to provide an output method and unit in which a host intending to transmit output data is identified, it is checked whether the host is the one which transmitted output data during the previous output, and offsetting is performed during output in accordance with the result of the check.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is a block construction diagram of a printer according to the present invention;

Fig. 2 is a flowchart showing main processing in the first embodiment;

Fig. 3 is a flowchart showing communication request check processing in the first embodiment;

Fig. 4 is a flowchart showing processing related to data entry in the first embodiment;

Fig. 5 is a flowchart showing printing in the first embodiment;

Fig. 6 shows a cross-sectional structure of a laser beam printer in an embodiment;

Fig. 7 is a flowchart showing communication request check processing in the third embodiment;

Fig. 8 is a block diagram showing a construction of a printer of a typical embodiment of the present invention;

Fig. 9 shows the details of a structure of a printer 10;

Fig. 10 shows correspondences between host IDs of host computers connected to a printer and types of emulation; and

Fig. 11 is a flowchart showing a procedure in which hosts serving as suppliers of print data are discriminated and printout is performed.

DESCRIPTION OF THE PREFERRED EMBODIMENTS

Embodiments relating to the present invention will be described in detail with reference to the appended drawings.

This embodiment will be described on the assumption that a laser beam printer based on an electrophotographic process using toner as a coloring material is employed as a printer.

Fig. 6 is a cross-sectional diagram showing an internal structure of a laser beam printer (hereafter, LBP) of an embodiment. The LBP can receive character patterns and form data from data sources which are not shown, and register them.

In Fig. 6, 1200 denotes an LBP body. The LBP body inputs and stores print information (for example, character codes), and form information or macroinstructions, which are supplied from host computers connected externally, creates associated character patterns or form patterns in accordance with the information, and forms images on recording paper that is a recording medium. 300 denotes an operation panel on which switches and LED indicators are arranged for operations. 1210 denotes a controller that controls the LBP 100 as a whole, and analyzes character information supplied from host computers. The controller 1210 converts character information into a video signal representing character patterns associated with the character information, and outputs the video signal to a laser driver 102.

The laser driver 102 is a circuit for driving a semiconductor laser 103. In response to an input video signal, the laser driver 102 turns on or off a laser beam 104 emitted from the semiconductor laser 103.

The laser beam 104 is swung right and left by a rotary polygon mirror 105 to scan an electrostatic drum 106. Thereby, the electrostatic drum 106 is exposed to light. Then, an electrostatic latent image consisting of character patterns is formed on the electrostatic drum 106. The latent image is developed by a developing unit 107 surrounding the electrostatic drum 106, then transferred to recording paper. Cut forms are employed as the recording paper. The cut forms of recording paper are accommodated in a paper cassette loaded in the LBP 100, fetched into the printer by means of a paper feed roller 109 and transport rollers 110 and 111, then supplied to the electrostatic drum 106.

Fig. 1 is a block construction diagram of a printer, which is an LBP, of an embodiment. The printer comprises host computers 1100a and 1100b, and a printer body 1200. In this embodiment, two host computers 1100a and 1100b are connected.

Host computers 1100a and 1100b send print data and control instructions for a printer body 1200 to the printer body 1200, and instructs the printer body 1200 to perform printing. The printer body 1200, as described previously, is divided into a controller 1210 and an engine 1220.

The controller 1210 receives print data and control instructions from the host computer 1100a or 1100b, creates bit map information according to the received data and instructions, then transfers the information as a video signal to the engine 1220. The engine 1220 receives the video signal from the controller 1210, and forms a visible image of the signal on paper. The controller 1210 communicates with the engine 1220, whereby the controller 1210 reports the state of the engine 1220 externally and controls the engine 1220 as a whole.

The controller 1210 will be described in more detail.

The controller 1210 comprises host interfaces 1211a and 1211b, a CPU 1212, a RAM 1213, a ROM 1214, a video interface 1215, an operation panel 1216, and a data bus 1217. The host interfaces 1211a and 1211b transfer data to or from the host computers 1100a and 1100b. In this embodiment, two host interfaces are installed. The CPU 1212 controls the printer as a whole, analyzes print data, and creates bit map information. The RAM 1213 is used to interpolate data sent from the host computers 1100a and 1100b, bit map information, or various registered data, and to temporarily store information when serving as a work area for information processing. The ROM 1214 contains programs for controlling the printer, which will be described later, and font information. The video interface 1215 transfers data to or from the engine 1220. The operation panel 1216 sets, modifies, or displays parameters for specifying print environments. Data transferred within the controller 1210 flows over the data bus 1217. Arrows in Fig. 1 indicates data flows.

Next, operation proc dur s of the CPU 1212 in the controller 1210 will be described in conjunction with Figs 2 to 5. Programs relating to the flowcharts reside in the ROM 1214.

Fig. 2 shows the processing for controlling a printer as a whole. The processing of Fig. 2 calls the processing of Figs. 3, 4, and 5 sequentially. The flowchart of Fig. 3 shows the processing for checking if a new communication request is issued for a host interface for which no data was sent when the processing was executed previously. Fig. 4 shows the processing for reading data from a designated host interface and writing the data in an input buffer for the interface. Fig. 5 shows the processing for reading data from the input buffer and printing the read data. The subsequent description will be made on the assumption that a variable "which_i_f" is reserved in the RAM 1213.

A specific example of processing will be described. For simple explanation, data is sent from the host computer 1100a, printing is performed on the data, and data is sent from the host computer 1100b during the printing.

When the power supply of the printer is turned on, first, the components of the printer are initialized at a step S21. The initialization of the step S21 includes initialization of, for example, a variable "which_i_f" reserved in the RAM 1213. The variable "which_i_f" contains the information which host interface is currently engaged in communication and in what order communication requests are issued.

At the next step S22, the processing shown in Fig. 3 is called.

When the processing is called, at a step S31, the information recorded in the variable "which_i_f" is checked to see if a communication request is issued from a host computer for a host interface which did not communicate with the printer previously. If a new communication request is not issued, the processing terminates. If a new communication request is issued, the processing passes to a step S32. In this embodiment, first of all, data is sent from the host computer 1100a. Therefore, the processing passes to the step S32.

At the step S32, the information via which host interface communication must be done is appended to the variable "which_i_f". Then, the processing terminates. In this embodiment, the information that the host interface 1211a is the host interface via which communication must be done is appended to the variable "which_i_f". Needless to say, when the host interface 1211b is the host interface via which communication must be done, data indicating the fact is appended. When the processing of Fig. 3 terminates, the main processing of Fig. 2 passes to a step S23.

At the step S23 in Fig. 2, based on the information recorded in the variable "which_i_f", the processing of Fig. 4 is started for a host interface via which communication must be done. In short, the process-

ing of Fig. 4 is called.

At a step S41 in Fig. 4, data is read via the designated host interface, and the data is written in an input buffer (reserved in th RAM 1213) associated with the host interface. Then, the processing of Fig. 4 terminates. In this embodiment, data is read via the host interface 1211a. In this embodiment, even if data is sent sequentially via the host interface 1211a, after data is read for a certain period of time or a certain amount of data is read, communication is suspended to terminate the processing of Fig. 4. Owing to this control procedure, a consecutive data flow can be read by receiving inputs via multiple host interfaces alternately. When the processing of Fig. 4 terminates as mentioned above, processing passes to a step S24 in Fig. 2.

At the step S24 in Fig. 2, the processing of Fig. 5 is called.

At a step S51 in Fig. 5, the information recorded in the variable "which_i_f" is checked to see for which one of host interfaces currently engaged in communication a communication request is issued first. Then, the processing passes to a step S52. In this embodiment, the host interface 1211a is the host interface for which a communication request is issued first.

At the step S52, it is checked if data resides in the input buffer for the host interface that was recognized at the step S51 as a host interface for which a communication request is issued first. If no data resides, the processing passes to a step S53. If data resides, the processing passes to a step S54.

At the step S54, data is read from the input buffer for the host interface selected at the step S52. The data is analyzed to perform necessary processing. Then, the processing of Fig. 5 is terminated. What is referred to as necessary processing is that data sent from the host computer is used to develop bit map information or that bit map information is printed on paper.

On the other hand, at the step S53, on the assumption that communication via the host interface, which was recognized at the step S52 as a host interface for which a communication request had been issued first, has terminated, the recorded contents of the variable "which_i_f" are updated. Then, the processing of Fig. 5 is terminated. To be more specific, during the update, the name of the host interface selected at the step S52 is deleted from the list of host interfaces currently engaged in communication, and the information of the remaining host interfaces whose orders in which communication requests were issued are incremented by one is recorded in the variable "which_i_f".

When the processing of Fig. 5 terminates as described above, processing returns to the step S22 in Fig. 2.

When the processing from the step S22 to S24 in Fig. 2 is repeated, data are read from the host computers repeatedly, and the data are printed continuously. In this embodiment, at this stage of the description, data have been read from the host computer 1100a via the host interface 1211a. Thereafter, it will be described what kind of processing will be executed when a communication request is issued by the host computer b 1100b during the aforesaid communication with the host computer a 1100a.

Within a sequence of operations from the steps S22 to S24, the printer is communicating with the host computer 1100a and the communicated data are being processed. In this sequence, if the host computer 1100b issues a communication request, it is detected at a step S31 in Fig. 3 that a new communication request has been issued. At a step S32, the recorded contents of the variable "which_i_f" are updated. Specifically, such information that the host interface 1211b also gets engaged in communication is appended to the variable "which_i_f" recording the information that only the host interface 1211a is engaged in communication at this stage.

Next, data is read from the host computer 1100a via the host interface 1211a at the step S23 in Fig. 2. Then, data is read from the host computer 1100b via the host interface 1211b.

Then, the processing from the step S22 to S24 in Fig. 2 is repeated. Thereby, while data concerning the usage of the data from the host computer 1100a and of the data from the host computer 1100b is being read, printing based on the data sent from the host computer 1100a can be continued.

When all the data from the host computer 1100a have been processed, the input buffer for the host interface 1211a becomes empty. Thereby, processing passes from a step S52 to a step S53 in Fig. 5. The information concerning the host interface 1211a is deleted from the recorded contents of the variable "which_i_f". Thereafter, the host computer 1100b is recognized as a host computer that had issued the first communication request. Then, printing is continued using data read from the host computer 1100b.

As described above, even when one of two host interfaces is used to communicate with a host computer, a host interface, which is not used for communication, can function on receipt of data. Owing to this control method, even when one of two host interfaces is used to communicate with a host computer, data can be sent from a host computer connected to the other host interface. After printing based on data sent from a host computer that started communication earlier is completed, printing based on data sent from the other host computer can be started.

[Description of the second embodiment]

In the aforesaid first embodiment, two host interfaces are installed. The present invention can apply to

the same processing to be performed when three host interfaces are installed.

[Description of the third embodiment]

In the aforesaid first and-second embodiments, print data sent from multiple host computers can be received. Needless to say, what is actually printed is a series of print data sent from a single host computer. Printing of data received from other host computers is not carried out until the previous printing is completed. Therefore, it is uncertain whether the printer has received the data sent from other hoot computers. In the third embodiment, it is notified externally from which host computer data is being received.

To realize the notification, the previous routine shown in Fig. 3 is replaced with the one shown in Fig. 7 in the third embodiment.

Steps S31 and S32 in Fig. 7 are identical to the corresponding ones in Fig. 3. Only a step S33 in Fig. 7 will be described.

When processing has passed to the step S33, the data indicating which host computers are currently communicating with the printer is specified in the variable "which_i_f". Therefore, at the step S33, the information corresponding to the data specified in the variable "which_i_f" is displayed on an operation panel 1216. For example, LEDs are installed on the operation panel 1216 by the number of host computers connectable to the printer. Then, an LED associated with a host computer that is communicating with the printer is lit. As described in the first embodiment, however, for a printer in which only two host computers can be connected, only one LED is required. Alternatively, when the operation panel 1216 has a liquid crystal display (hereafter, LCD), a message or symbol indicating a host computer currently communicating with the printer may be displayed.

Furthermore, not only which host computer is currently communicating with the printer but also from which host computer print data being printed originate may be notified. In this case, LEDs may be installed in association with host computers, or a message or symbol may be displayed on an LCD.

In the first to third embodiments, no comment is made on a type of host interface. This is because the present invention can apply to any type of host interface.

In the previous embodiments, an electrophotographic printing type engine is employed as an engine for a printer The present invention can apply to any printer having any type of engine; such as, an engine of a thermal recording type, an ink jet recording type, or a wire dot recording type.

In the embodiments, data received from host computers are stored in buffers, which are reserved in an RAM 1213, in one-to-one correspondence with the host computers. A hard disk or other external memory may be installed so that the data can be stored in the external memory.

Mor over, in the embodiments, while printing is under way, it is determined whether a transmission request is issued for any other host interface. Every time data is received via an interface, an interrupt signal may be fed to a CPU 1212. In response to each interrupt signal, the CPU 1212 may execute an interrupt routine to update the variable "which_i_f".

As described so far, according to the present invention, print data can be received from multiple data generation sources. Therefore, even when printing is executed for a specific data generation source, other data generation sources need not delay outputting their print data.

With which data generation source the printer is communicating is notified externally. Therefore, the users of respective data generation sources can confirm that the user-specified printing is being carried out.

[Offset stacking]

Other preferred embodiments of the present invention will be described with reference to the appended drawings.

[Outline of the printer (Fig. 8)]

Fig. 8 is a block diagram showing a construction of a printer representing a typical embodiment of the present invention. In Fig. 8, 1a to 1n denote hosts for transferring data. 2 denotes an SCSI interface. 3 denotes an input buffer that stores input data temporarily. 4 denotes a CPU that controls the printer as a whole. 5 denotes a ROM containing control programs and various processing programs. 6 denotes a RAM used to store various information necessary for printing or used as a work area of each program. 7 denotes a pattern memory containing character patterns in association with character codes. 8 denotes an image memory that stores images for one page. 9 denotes a printer interface that extends control throughout a printer, for example, transfers image data existent in the image memory to a printer. 10 denotes a printer that actually performs printing.

The CPU 4 analyzes data read from the input buffer 3, and develops images of print data in the image memory 8 while referencing character patterns existent in the pattern memory 7.

[Detailed description of a construction of a printer 10 (Fig. 9)]

Fig. 9 shows an internal structure of a printer 10 (hereafter, LBP) capable of registering character patterns received from a data source (not shown) or form data, and outputting image data provided via the prin-

ter interface 9 on recording paper. In Fig. 9, 100 denotes an LBP body, wherein character information (character codes), and form information or macroinstructions, which are supplied by hosts 1a to 1n, are input and stored, associated character and form patterns are created in accordance with the information, and images are formed on recording paper that is a recording medium. 101 denotes an operation panel on which switches and LEDs, which are designed for operation, and an LCD are arranged. 1000 denotes a print controller that controls the LBP 100 as a whole and analyzes character information supplied by external units. The print controller 1000 converts character information into a video signal representing character patterns associated with the character information, and outputs the video signal to a laser driver 102.

The print controller 1000 accommodates, as shown in Fig. 8, an SCSI interface 2, an input buffer 3, a CPU 4, a ROM 5, a RAM 6, a character pattern memory 7, an image memory 8, and a printer interface 9.

The laser driver 102 is a circuit for driving a semiconductor laser 103. In response to an input video signal, the laser driver 102 turns on or off a laser beam 104 emitted from the semiconductor laser 103. The laser beam 104 is deflected laterally by a rotary polygon mirror 105 to scan an electrostatic drum 106. Thereby, an electrostatic latent image consisting of character patterns is formed on the electrostatic drum 106. The latent image is developed by a developing unit 107 surrounding the electrostatic drum 106, then transferred onto recording paper. Cut forms are employed as the recording paper. The cut forms of recording paper are accommodated in a paper cassette loaded in the LBP 100, taken into the printer by means of a paper feed roller 109 and transport rollers 110 and 111, then fed to the electrostatic drum 106.

[Description of printer operations (Figs. 10 and 11)]

Fig. 10 shows correspondences between hosts connected to a printer of this embodiment and types of emulation. This information is stored in a RAM 6. In Fig. 10, 201 denotes an ID number of a host (hereafter, host ID). 202 denotes a compatible type of emulation. 203 indicates that a host ID is 7. For the host having a host ID 7, an emulation program (emulation 1) specified in 204 is executed to analyze data. Similarly, for a host having a host ID 6, an emulation program (emulation 2) specified in 206 is executed to analyze instructions and to process data.

Operations executed by a printer having the foregoing construction will be described with reference to the flowchart of Fig. 11. Herein, any of the hosts 1a to 1n transmits data, and the printer of this embodiment starts printing. When communicating with the printer, before transferring data, the hosts informs the printer of their host IDs.

In the above printing, a host ID sent from a host is identified. Every time hosts transmitting print data are changed, an offset flag residing in the RAM 6 is turned on or off. Based on the value of the offset flag, it is controlled whether or not to offset an output of the printer.

First, at a step S1, a host ID of a host that has called the printer is checked. At a step S2, the information of correspondences between connected hosts and emulation types, which is stored in the RAM 6, is referenced to select an emulation program (hereafter, emulator) associated with the same ID as the host ID checked at the step S1. The selected emulator is executed to analyze input data. At the step S3, a host ID (old ID) of a host, which happened to be a data supplier for the previous printing, is compared with the host ID (new ID) of the host, which has called the printer at the step S1. When the old and new host IDs are different from each other (hosts supplying print data are changed), the offset flag is turned on. Then, data is fetched from the input buffer 3 at a step S4.

At a step S5, it is checked if the host has completed data transmission. If it is determined that data has come to an end, processing passes to a step S6. Then, the new ID is set as an old ID. By contrast, when it is determined that data transmission has not terminated, processing passes to a step S7. The CPU analyses transmitted data. If the data is a control code, an instruction is analyzed. If the data is a character code, a font pattern associated with the character code is read from the pattern memory 7 and written in the image memory 8. Next, at a step S8, an amount of output data supplied to the image memory 8 is checked to see if form feed is required. If it is determined that form feed is required, processing passes to a step S9. Then, the input buffer 3 is closed. On the contrary, if form feed is not required, processing returns to the step S4. Then, data entry is repeated.

At a step S10, the offset flag is checked. If the offset flag value is on, processing passes to a step S11. Then, offset is specified in the printer 10 via the printer interface 9. By contrast, if the offset flag value is off, processing passes to a step S12. Then, data developed in the image memory 8 is transferred to the printer 10 via the printer interface 9, and printed. At this time, if the offset flag value is off, outputs of the printer are not offset. If the offset flag value is on, outputs are offset by a specified amount.

As described above, according to the present embodiment, every time hosts serving as suppliers of print data are changed, whether or not print outputs are offset is switched, then printout is carried out. Therefore, printouts can be easily sorted by hosts.

In this embodiment, when data is transmitted from multiple host computers, offset stacking is carried out. The present invention is not limited to this

mode. For example, offset stacking may be performed depending on the type of emulation. When a host ID or equivalent is not assigned, standard emulation may be pre-programmed. Alternatively, hosts may be discriminated by outputting partition paper at every start of prin(out of data from a different host.

The present invention may apply to a printing system made up of multiple equipment or a printer formed with a single equipment. The present invention can, needless to say, apply to a printing system or a printer in which programs are implemented.

As described so far, according to the present invention, a host transmitting print data is identified, and checked if the host is identical to the one that transmitted print data during the previous printing. Then, based on the result of the comparison, print outputs are offset. Every time hosts transmitting print data are changed, offsetting is performed alternately. Consequently, printouts can be easily sorted by hosts.

Claims

1. An output unit, comprising:
   a plurality of interfaces for connecting a plurality of data generation sources;
   determining means for determining for each of said interfaces whether a reception request has been issued therefor;
   storage means for inputting data via at least one of said interfaces for which a reception request has been issued and storing the input data in one-to-one correspondence with respective interfaces; and
   control means for controlling said output unit so that after output of a first received data, stored in said storage means, has been completed, output of a second received data is started.

2. An output unit according to claim 1, further comprising informing means for externally informing the states of at least one of said interfaces for which a reception request has been issued or the states of at least one of said interfaces concerned with said storage means.

3. An output method, comprising the steps of:
   determining, for each of a plurality of interfaces connected to a plurality of data generation sources, whether a reception request has been issued therefor;
   inputting data via at least one of the interfaces for which a reception request has been issued and storing the input data in one-to-one correspondence with respective interfaces; and
   controlling output processing so that after output of received data, firstly stored in said inputting step, has been completed, output of received data, secondly stored in said inputting step, is started.

4. An output method according to claim 3, further comprising the step of informing externally the states of at least one of the interfaces for which a reception request has been issued or the states of at least one of the interfaces concerned with said inputting step.

5. A printer that receives print data, records images on specified recording medium, and outputs the images, comprising:
   a plurality of interfaces for connecting a plurality of data generation sources;
   determining means for determining for each of said interfaces whether a reception request has been issued therefor;
   storage means for inputting print data via at least one of said interfaces for which a reception request has been issued and storing the input print data in one-to-one correspondence with respective interfaces; and
   control means for controlling said printer so that after printing of a first received data, stored in said storage means, has been completed, printing of a second received data is started.

6. An output method, comprising:
   a discrimination process in which a host intending to transmit data is discriminated from a plurality of hosts;
   a preparation process in which preparations are made to receive data from the host intending to transmit data in accordance with the result of discrimination made in said discrimination process;
   a comparison process in which it is checked if a host that transmitted output data during the previous output is identical to the host identified in said discrimination process; and
   a control process in which offsetting is controlled so that outputs of data will be offset in accordance with the result of comparison made in said comparison process.

7. An output method according to claim 6, wherein a host intending to transmit output data is discriminated in said discrimination process by checking ID numbers assigned uniquely to said plurality of hosts.

8. An output unit, comprising:
   a discriminating means that identifies a host intending to transmit data from a plurality of hosts;
   a preparing means that makes preparations to receive from said host intending to trans-

mit data in accordance with the result of discrimination made by said discriminating means;

a comparing means that checks if a host which transmitted data during the previous output is identical to the host identified by said identifying means; and

a control means that controls offsetting so that outputs of data will be offset in accordance with the result of comparison made by said comparing means.

9. An output unit according to claim 6, wherein said preparing means includes emulation means in association with said plurality of hosts so that data can be received or output from or to said plurality of hosts.

10. A printer having a plurality of interfaces which can be connected to individual data sources to receive data therefrom, and measuring means for storing data received from the data sources so that whilst the printer is printing data received from one data source the memory means can store data received from other data sources through the associated interfaces.

European Patent Office

EUROPEAN SEARCH REPORT

Application Number
EP 93 30 0918

DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int. Cl.5)
X	US-A-5 018 079 (SHUKUNAMI, ET AL) * the whole document *	1-10	G03G15/00 H04N1/21 G06F3/12 G06K15/00
X	WO-A-9 012 359 (EASTMAN KODAK) * the whole document *	1-10	
A	US-A-4 965 771 (MORIKAWA, ET AL) * the whole document *	1-10	
A	US-A-5 047 957 (IKENOUE) * the whole document *	1-10	
A	EP-A-0 468 762 (HITACHI) * the whole document *	1-10	
A	US-A-4 864 417 (WATANABE, ET AL) * the whole document *	1-10	
A	EP-A-0 450 663 (CANON) * the whole document *	1-10	

TECHNICAL FIELDS SEARCHED (Int. Cl.5)

G03G
H04N
G06F
G06K

The present search report has been drawn up for all claims

Place of search	Date of completion of the search	Examiner
BERLIN	21 JUNE 1993	HOPPE H.

CATEGORY OF CITED DOCUMENTS

X : particularly relevant if taken alone
Y : particularly relevant if combined with another document of the same category
A : technological background
O : non-written disclosure
P : intermediate document

T : theory or principle underlying the invention
E : earlier patent document, but published on, or after the filing date
D : document cited in the application
L : document cited for other reasons

& : member of the same patent family, corresponding document

EPO FORM 1503 03.82 (P0401)

FIG. 1

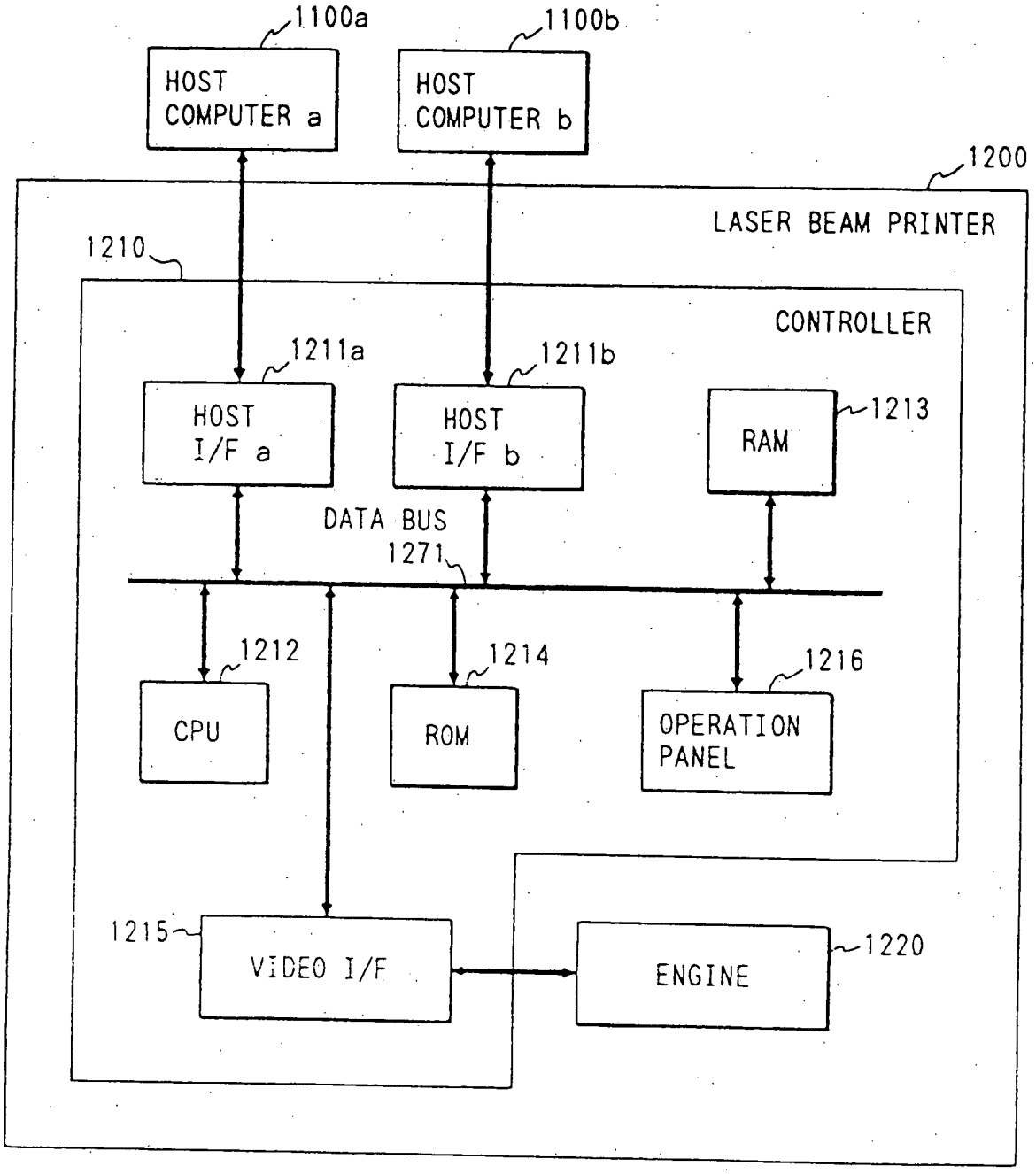

1100a
1100b
HOST COMPUTER a
HOST COMPUTER b
1200
LASER BEAM PRINTER
1210
CONTROLLER
1211a
1211b
HOST I/F a
HOST I/F b
RAM
1213
DATA BUS
1271
1212
1214
1216
CPU
ROM
OPERATION PANEL
1215
VIDEO I/F
ENGINE
1220

FIG. 2

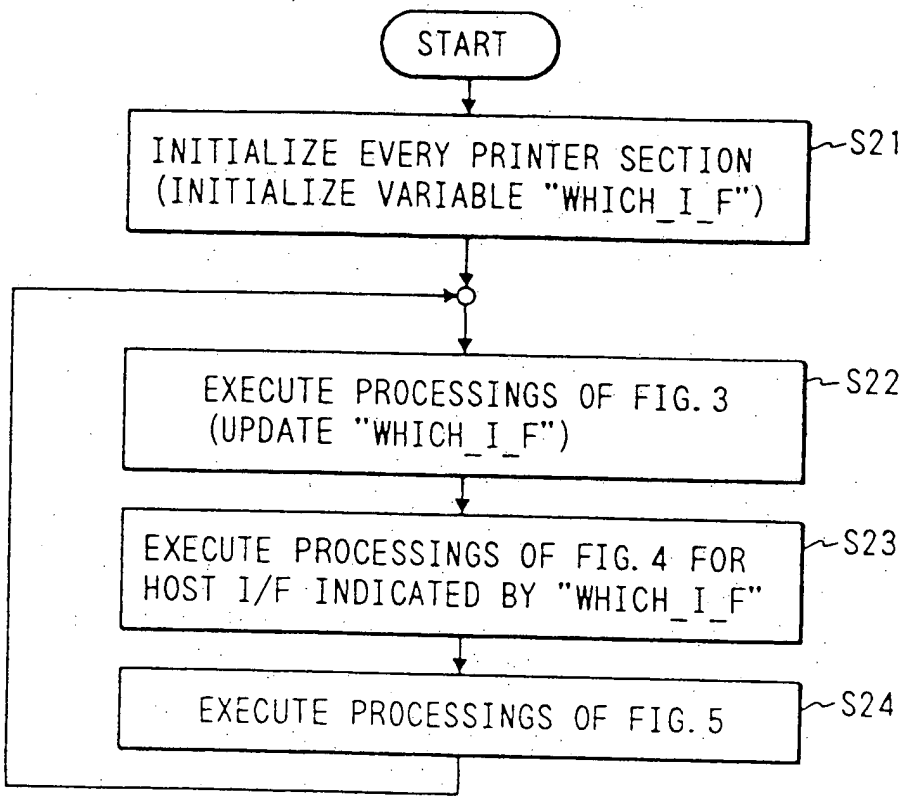

START
INITIALIZE EVERY PRINTER SECTION (INITIALIZE VARIABLE "WHICH_I_F")
S21
EXECUTE PROCESSINGS OF FIG. 3 (UPDATE "WHICH_I_F")
S22
EXECUTE PROCESSINGS OF FIG. 4 FOR HOST I/F INDICATED BY "WHICH_I_F"
S23
EXECUTE PROCESSINGS OF FIG. 5
S24

FIG. 3

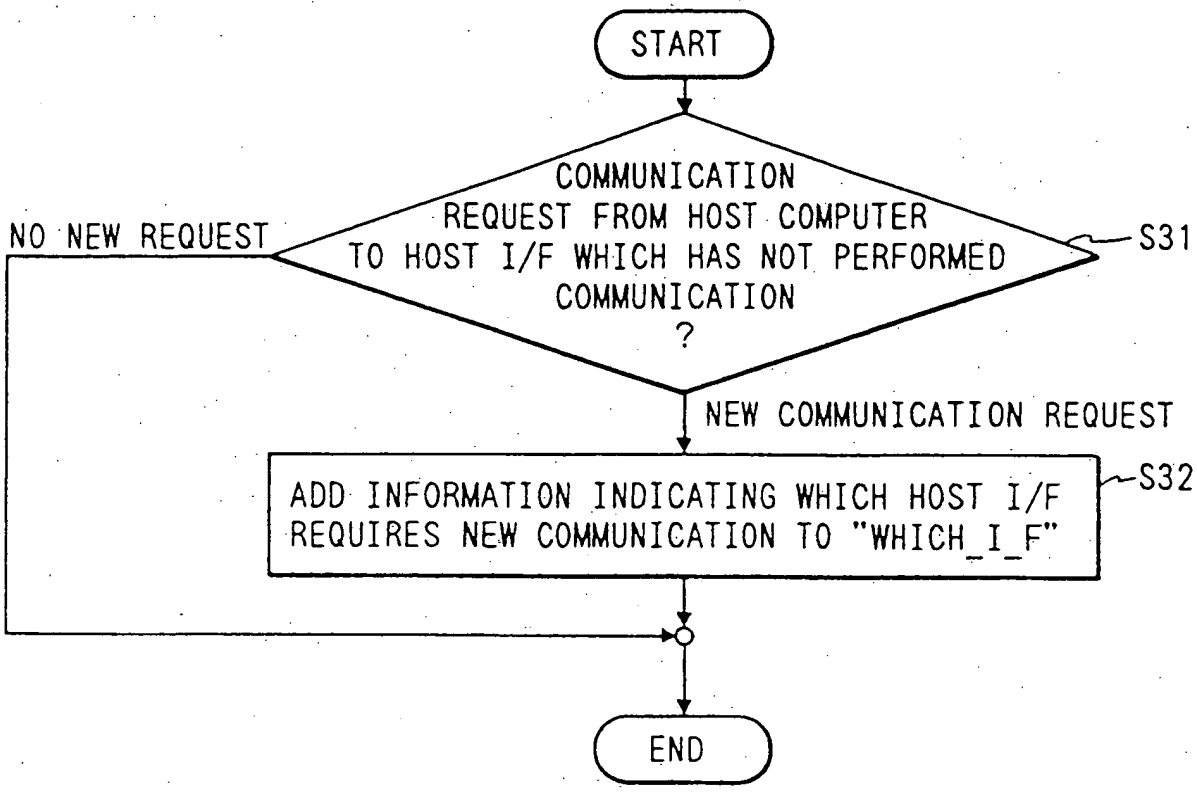

START
COMMUNICATION REQUEST FROM HOST COMPUTER TO HOST I/F WHICH HAS NOT PERFORMED COMMUNICATION ?
S31
NO NEW REQUEST
NEW COMMUNICATION REQUEST
ADD INFORMATION INDICATING WHICH HOST I/F REQUIRES NEW COMMUNICATION TO "WHICH_I_F"
S32
END

FIG. 4

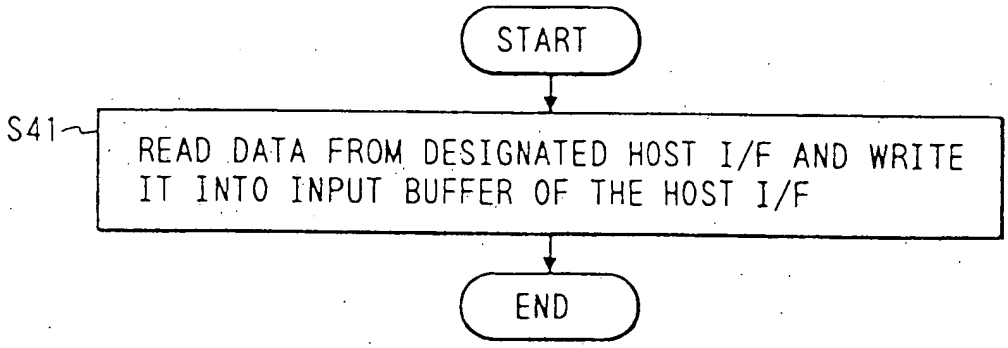

START
S41
READ DATA FROM DESIGNATED HOST I/F AND WRITE IT INTO INPUT BUFFER OF THE HOST I/F
END

FIG. 5

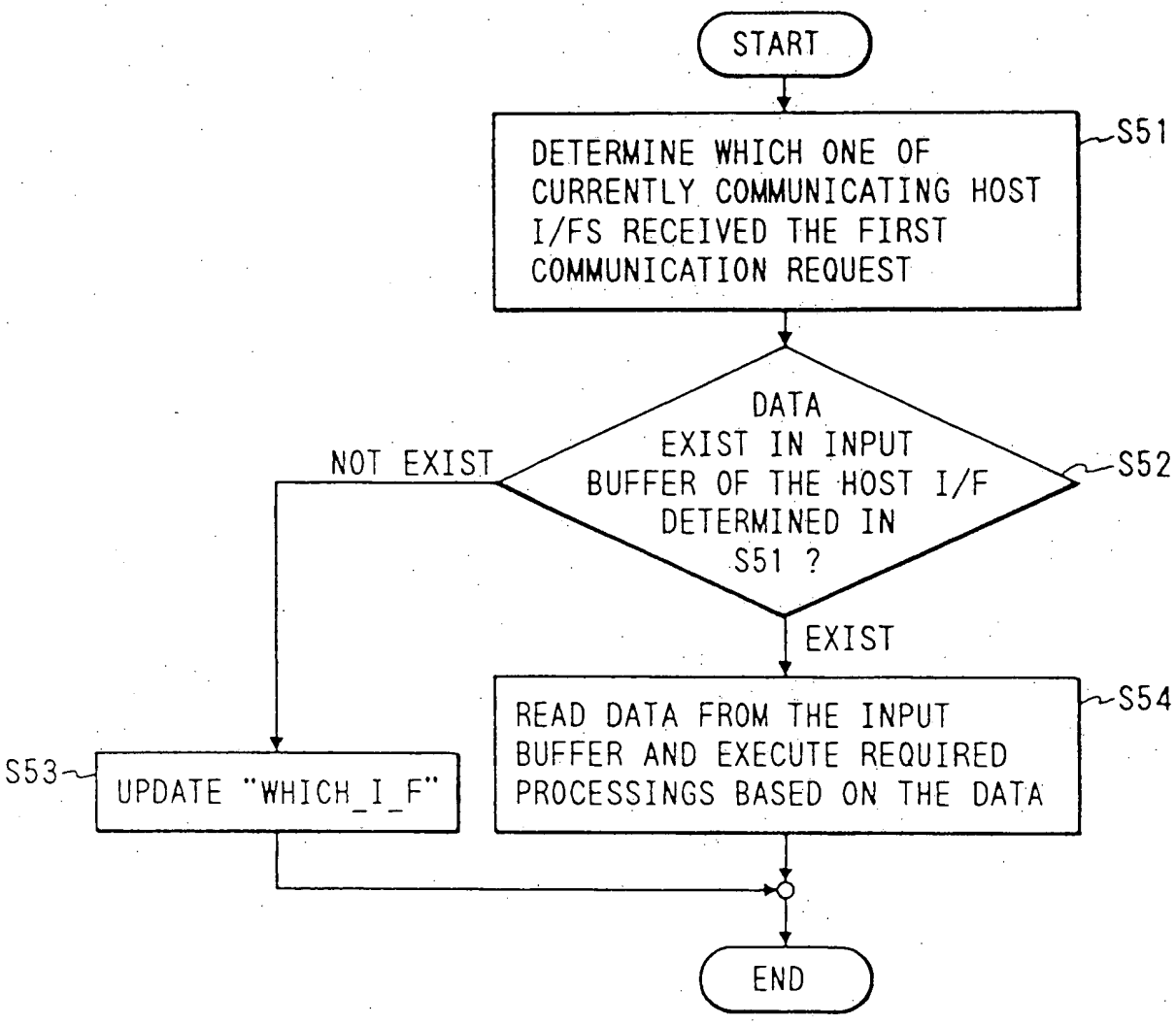

START
DETERMINE WHICH ONE OF CURRENTLY COMMUNICATING HOST I/FS RECEIVED THE FIRST COMMUNICATION REQUEST
S51
DATA EXIST IN INPUT BUFFER OF THE HOST I/F DETERMINED IN S51 ?
S52
NOT EXIST
EXIST
READ DATA FROM THE INPUT BUFFER AND EXECUTE REQUIRED PROCESSINGS BASED ON THE DATA
S54
S53
UPDATE "WHICH_I_F"
END

FIG. 6

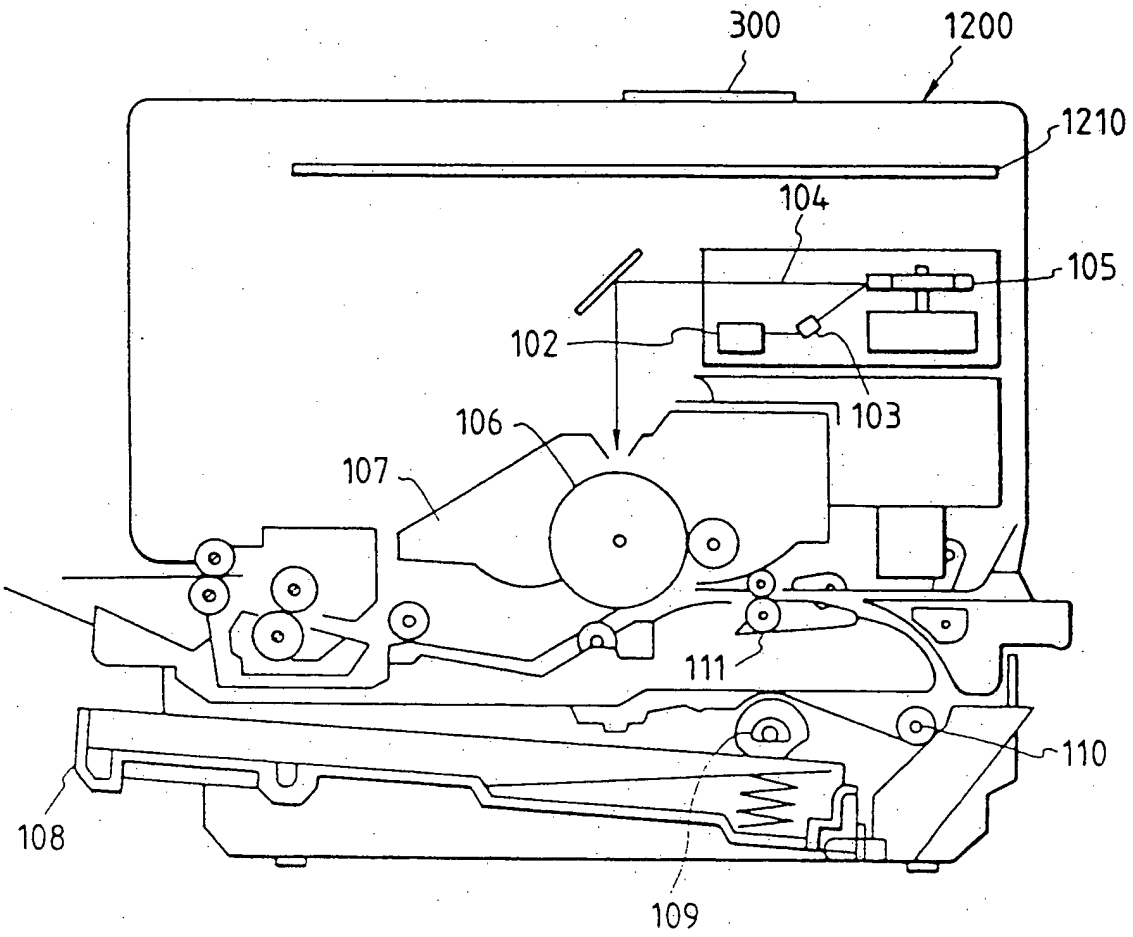


FIG. 7

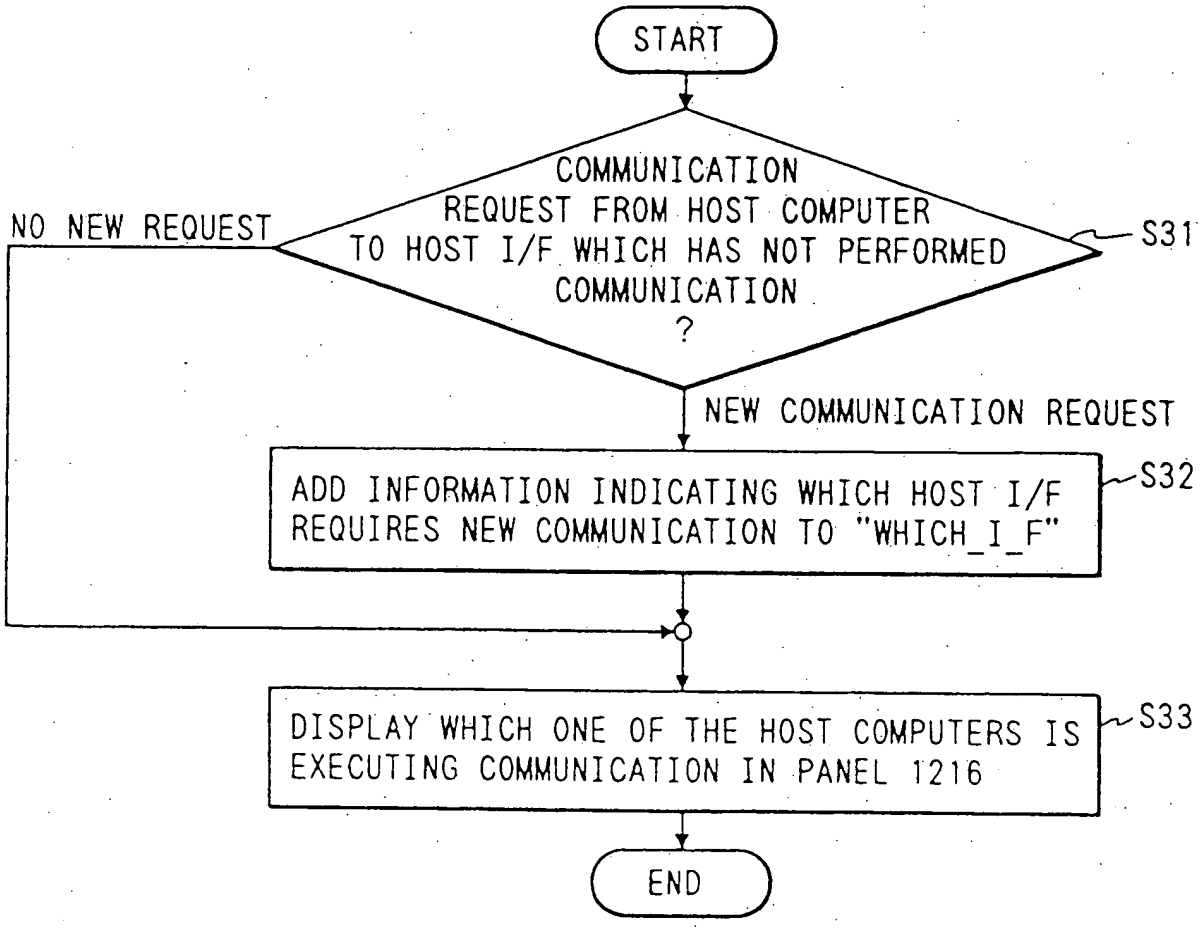

START
COMMUNICATION REQUEST FROM HOST COMPUTER TO HOST I/F WHICH HAS NOT PERFORMED COMMUNICATION ?
S31
NO NEW REQUEST
NEW COMMUNICATION REQUEST
ADD INFORMATION INDICATING WHICH HOST I/F REQUIRES NEW COMMUNICATION TO "WHICH_I_F"
S32
DISPLAY WHICH ONE OF THE HOST COMPUTERS IS EXECUTING COMMUNICATION IN PANEL 1216
S33
END

FIG. 8

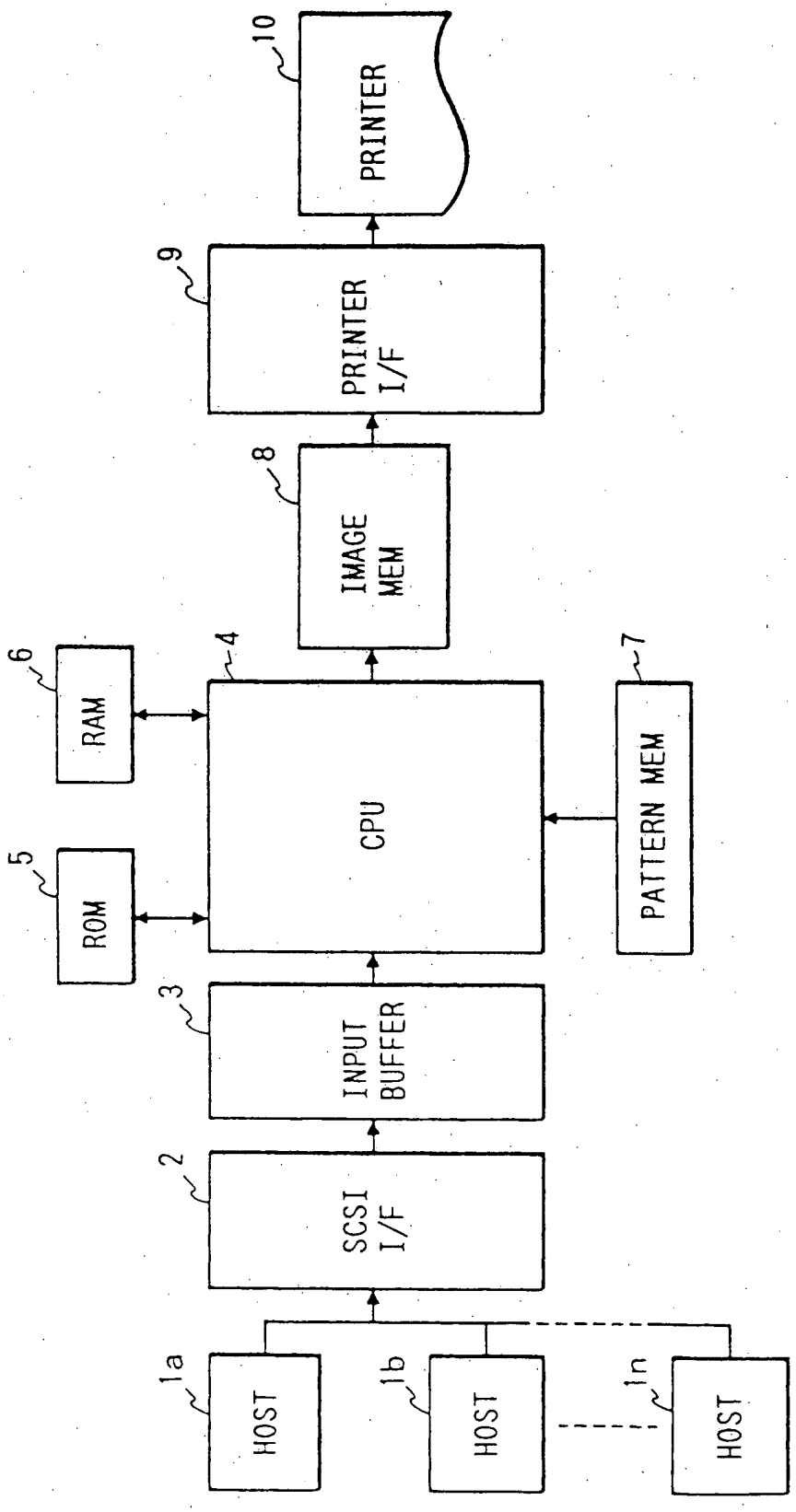

1a
HOST
1b
HOST
1n
HOST
2
SCSI I/F
3
INPUT BUFFER
4
CPU
5
ROM
6
RAM
7
PATTERN MEM
8
IMAGE MEM
9
PRINTER I/F
10
PRINTER

FIG. 9

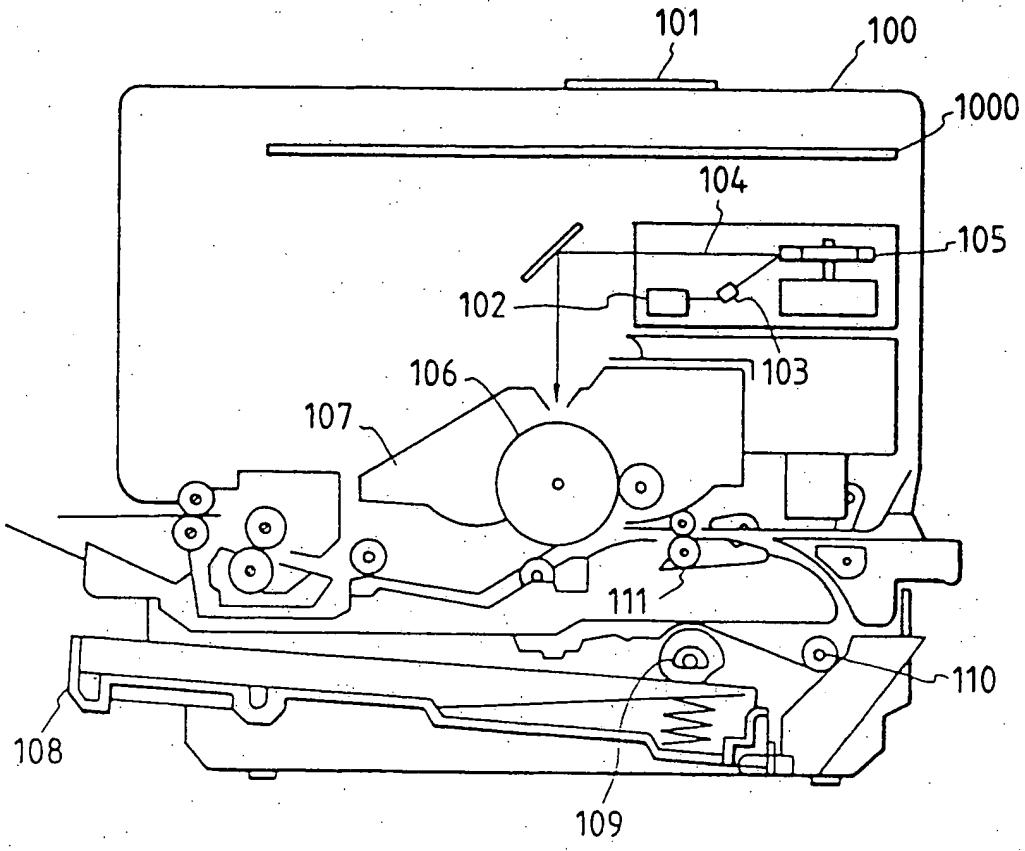
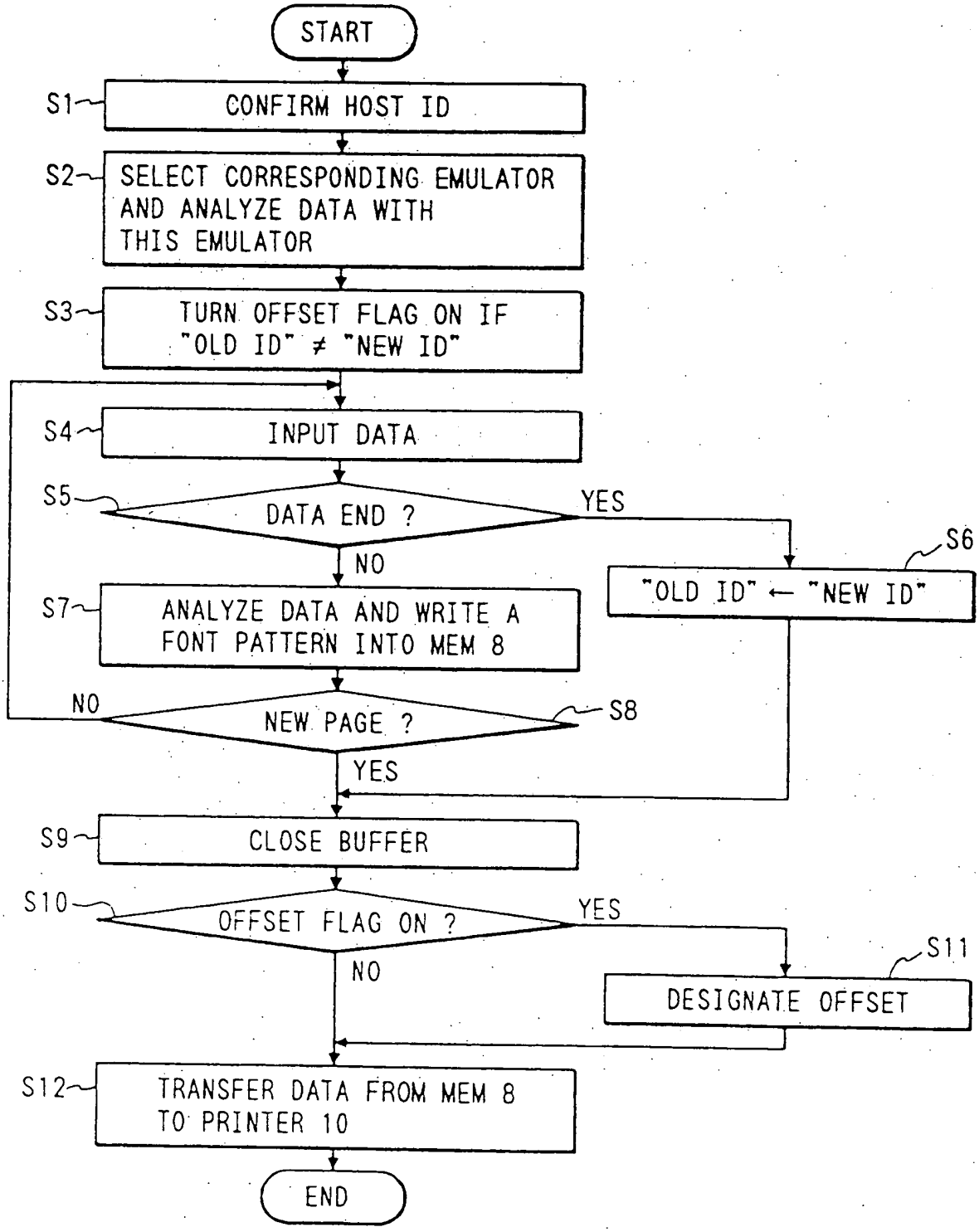


FIG. 10

201 HOST ID	202 EMULATION TYPE
203 7	EMULATION 1 204
205 6	EMULATION 2 206

FIG. 11


START
S1
CONFIRM HOST ID
S2
SELECT CORRESPONDING EMULATOR AND ANALYZE DATA WITH THIS EMULATOR
S3
TURN OFFSET FLAG ON IF "OLD ID" ≠ "NEW ID"
S4
INPUT DATA
S5
DATA END ?
YES
NO
S6
"OLD ID" ← "NEW ID"
S7
ANALYZE DATA AND WRITE A FONT PATTERN INTO MEM 8
NO
NEW PAGE ?
S8
YES
S9
CLOSE BUFFER
S10
OFFSET FLAG ON ?
YES
NO
S11
DESIGNATE OFFSET
S12
TRANSFER DATA FROM MEM 8 TO PRINTER 10
END